УДК 681.3.01

Н.Б. Шаховська, О.А. Басистюк
Національний університет «Львівська політехніка»
вул. С. Бандери, 32, м. Львів, 79013

РОЗПІЗНАВАННЯ ОБЛИЧЧЯ ЗА ДОПОМОГОЮ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ

N.B. Shakhovska, O.A. Basystiuk
Lviv polytechnic national university, Ukraine
32, S. Bandery st., Lviv, 79013

FACE RECOGNITION BASED ON MACHINE LEARNING ALGORITHMS

У цій статті досліджуються алгоритми розпізнавання облич, які побудовані на машинному навчанні, наявні у них недоліки і розглядаються методи їх покращення. Зокрема запропоновано використовувати технологію гістограми орієнтованих градієнтів, L2-нормування і оцінки орієнтирів обличчя для досягнення найкращого результату розпізнавання.

Ключові слова: розпізнавання облич, машинне навчання, Гістограма орієнтованих градієнтів, оцінка орієнтирів обличчя, Python, dlib.

This paper explores machine learning algorithms for face recognition, there pro and cons and investigate methods of their improvement. In paper is proposed usage of histogram of oriented gradients(HOG), L2-norm and face landmarks estimation for achieving most accurate result of face recognition.

Keywords: face recognition, machine learning, Histogram of oriented gradients, face landmarks estimation, Python, dblib;

Вступ

Прогнозування ціни на будинок, пошук асоціативних залежностей, генерація нових даних на основі існуючих– усі ці проблеми можуть бути вирішені з використанням одного алгоритму машинного навчання чи ансамблю подібних. Проте, проблема розпізнавання облич є комплексною та складається з таких частин [1]:

1. Знайти людину та її обличчя рисунку;
2. Перебрати всі розпізнані обличчя, щоб визначити, чи це обличчя не є спотворене, наприклад: повернуте в бік, погано освітлене, частково перекрите іншими об’єктами. У разі наявності таких спотворень провести коректування даних.
3. Вибрати унікальні особливості обличчя, на які опиратиметься алгоритм під час порівняння, наприклад: довжина і ширина обличчя, наскільки великі очі або ніс і тп.
4. Порівнювати особливості обличчя з наявними даними для розпізнавання чи оригіналом зображення.

Мозок людини проводить ці операції автоматично і миттєво. Поки що, комп’ютери не здатні до такого рівня узагальнення даних, отже перед нами стоїть завдання навчити їх робити ці кроки окремо. Тобто метою роботи є розроблення ланцюжок з декількох алгоритмів машинного навчання.

Пошук усіх облич на картинці

Перший крок у нашій системі – це знаходження облич.

Розпізнавання облич є однією з функцій, вбудованих у камери сучасних засобів. Коли камера сама вміє автоматично вибирати обличчя, вона може переконатися чи всі вони у фокусі, чи усміхаються люди на фото і тому подібні речі. Проте, ми використовуватимемо це для інших цілей – знаходження місця розташування облич на фотографії.

Розпізнавання облич почало активно розвиватися з 2000-их, коли Paul Viola(Пауль Віола) і Michael Jones(Мішель Джонс) винайшли спосіб розпізнавання облич, який був достатньо швидкий для роботи навіть на дешевих камерах. Однак, зараз існує кращий спосіб, який був винайдений в 2005, Navneet Dalal(Навніт Далал) і Bill Triggs(Білом Тріггс), який називається Гістограма орієнтованих градієнтів (Histogram of Oriented Gradients, HOG) [2]. Принцип роботу алгоритму пошуку гістограми орієнтованих градієнтів полягає у виконанні таких кроків:

1. Розрахунок значення градієнта. Найпоширенішим способом розрахунку є використання одновимірної центрованої маски, переважно в горизонтальному і вертикальному напрямках (зустрічається використання в одному) з подальшим фільтруванням кольору або насиченості зображення з використанням таких фільтрувальних ядр:
   a. $[-1, 0, 1]$ і $[-1, 0, 1]^{\top}$
2. Побудова гістограмних комірок. Кожний піксель у комірці бере участь у голосуванні для каналів гістограми напрямків. Комірки можуть бути прямокутної чи круглої форми, канали гістограми рівномірно розподіляються від 0 до 180 або від 0 до 360 градусів, в залежності від того, вираховується “беззнаковий (unsigned)” чи “знаковий (signed) градієнт. Розробники алгоритму визначили, що для розпізнавання людини, якнайкраще підходить unsigned градієнт, використаний разом з 9 каналами гістограми.
3. Щоб визначити зміни освітленості та контрастності, сила градієнта повинна бути локально нормалізованою, що вимагає об'єднання комірки у більші блоки. Дескриптор HOG є тоді об'єднаним вектором компонентів нормалізованої гістограми комірок з усіх областей блоків. Ці блоки, як правило, перекриваються, що означає, що кожна комірка неодноразово вноситься до фінального дескриптора. Існують дві основні геометрії блоків: прямокутні блоки R-HOG і круглі блоки C-HOG. Блоки R-HOG, як правило, квадратні сітки, представлені трьома параметрами: кількість комірок на блок, кількість пікселів на комірку та кількість каналів на гістограму комірки. В експерименті з розпізнавання людей було встановлено, що оптимальними параметрами є чотири комірки 8x8 пікселів на блок (16x16 пікселів на блок) з 9 каналами гістограми.
4. Нормалізація. Було винайдено чотири методи нормалізації. Нехай V буде нашим ненормалізованим вектором, який містить усі гістограми в даному блоці, $\| V \|_k$ – його k-норма, при k = 1, 2 і e – якась мала константа(точне значення нам неважливе). Тоді коефіцієнт нормалізації можна розрахувати такими способами:

a. L2-norm: $f = \frac{V}{\|V\|_2^2 + e^2}$;

b. L2-hys: L2-norm обмежена (значення V > 0.2, прирівнюються до 0.2) і потім перенормовується

c. L1-norm: $f = \frac{V}{\|V\|_1 + e}$;

d. L1-sqrt: $f = \sqrt{\frac{V}{\|V\|_1 + e}}$;

e. Також було встановлено, що L1-norm дає гіршу продуктивність нормування порівняно з іншими трьома, проте будь-яке нормування значно підвищує ефективність розпізнавання.

5. Відправити нормалізований дескриптор в систему, базовану на контрольованому навчанні. Класифікатор векторної машини підтримки (SVM) - це бінарний

класифікатор, який шукає оптимальну гіперплощину. Після навчання на зображеннях, що містять певний об'єкт, в нашому випадку обличчя, класифікатор SVM зможе приймати рішення щодо наявності об'єкта на зображеннях.

Опрацювання зображення за допомогою HOG, є важливим кроком розпізнавання обличчя. Якщо ми аналізуємо пікселі напряму, темні та світлі зображення однієї і тієї ж людини матимуть абсолютно різні характеристики. А коли враховуватимемо тільки напрям зміни яскравості зображення, то як темні, так і яскраві зображення видаватимуть однаковий результат. Проте, збереження градієнта для кожного окремого пікселя дає нам занадто багато деталей. Для цього було вирішено розбивати зображення на квадрати 16x16 пікселів кожен. У кожному квадраті підрахуватимемо кількість векторів зміни градієнтів у кожному основному напрямку (вгору, вгору і вправо, просто вправо і тд). Після цього, ми замінимо цей квадрат 16x16 пікселів вектором (або умовно струлкою), яка найчастіше використовувалася в цьому квадраті.

Кінцевим результатом є перетворення оригінального зображення в зображення, яке має в собі основну структуру обличчя.

Щоб знайти обличчя в HOG представленні, треба просто порівняти його з вже відомим шаблоном формату HOG, який було отримано з навчальних зображень. Використовуючи цю техніку, тепер ми можемо легко знайти обличчя в будь-якому зображенні.

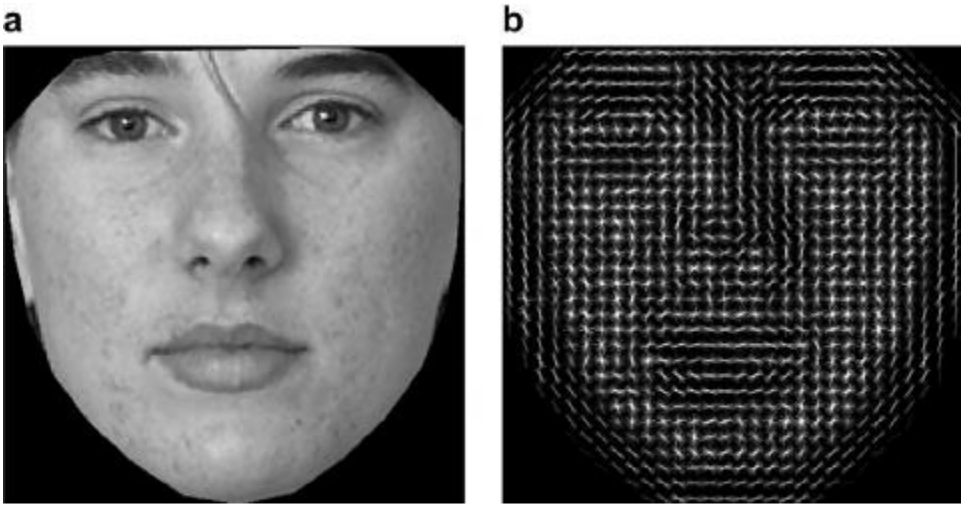


Рис. 1. А) Приклад обличчя В) Обличчя перетворене у HOG формат

HOG розпізнавання облич імплементоване програмно мовою Python з використанням бібілотеки dlib.

```
import sys
import dlib
from skimage import io

# Дістаємо назву зображення з масиву даних командного рядка
file_name = sys.argv[1]

# Створюємо розпізнавача облич за допомогою HOG, який є вбудований в бібліотеці dlib
face_detector = dlib.get_frontal_face_detector()
```



```
win = dlib.image_window()

# Завантажуємо зображення в масив
image = io.imread(file_name)

# Запускаємо HOG розпізнавача обличь для нашого зображення.
# Результатом будуть квадратики в яких знаходитимуться всі розпізнані обличчя
detected_faces = face_detector(image, 1)

print("I found {} faces in the file {}".format(len(detected_faces), file_name))

# Відкрити вікно для показу зображення
win.set_image(image)

# У циклом опрацьовуємо всі обличчя, які були знайдені на зображенні
for i, face_rect in enumerate(detected_faces):

    # Знайдене обличчя повертається, як набір координат всіх 4 сторін
    print("- Face #{} found at Left: {} Top: {} Right: {} Bottom: {}".format(i, face_rect.left(),
face_rect.top(), face_rect.right(), face_rect.bottom()))

    # Малюємо квадрат навколо кожного знайденого обличчя
    win.add_overlay(face_rect)

# Очікуємо натискання кнопки <Enter> для закриття вікна
dlib.hit_enter_to_continue()
```

Позиціонування обличчя

На минулому кроці ми отримали обриси обличчя, але зараз нам необхідно позиціонувати обличчя для розпізнавання, оскільки навіть трішки повернуте лице на зображенні – це вже дві принципово різні людини для комп'ютера.

Для для розв'язання цієї задачі використовуватимемо алгоритм, який називається оцінкою орієнтирів обличчя (face landmark estimation). Існує багато варіантів цього алгоритму, але у статті запропоновано використовувати спосіб, який був винайдений у 2014 році Вахід Каземі (Vahid Kazemi) та Джозефіна Суліван (Josephine Sullivan). Основна ідея полягає в наявності 68 орієнтирів (landmarks), які присутні на кожному обличчі – верхня частина підборіддя, внутрішній край брів, зовнішній край очей, нижня точка носу, також верхня і нижня точка губи і тд. Завдання полягає в тому, щоб навчити алгоритм шукати усі 68 орієнтирів на кожному обличчі, яке ми отримаємо після першого кроку.

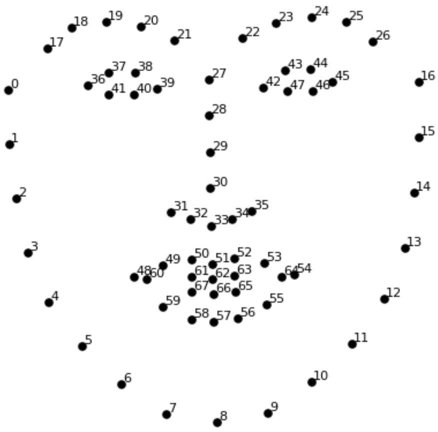


Рис. 2. Усі 68 точок-орієнтирів, які розпізнаватиме алгоритм на обличчі.

Отже, необхідно шукати усі ці орієнтири на картинці, а потім коректувати за допомогою простих маніпуляцій (повертання, збільшення або зменшення), аж до отримання найкращого результату, коли обличчя буде розташоване в центрі. У цьому способі не використовується жодне складне 3d перетворення, оскільки вони часто погіршують якість картинки, що ускладнює подальше розпізнавання.

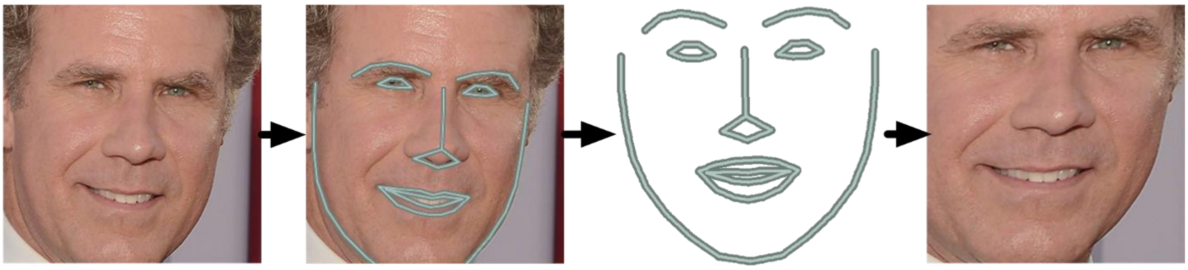


Рис. 3. Візуалізація того, як працює алгоритм пошуку орієнтирів лиця

Тепер для алгоритму розпізнавання не є проблемою повернуте на зображенні обличчя, оскільки це зображення можна опрацювати та відцентрувати. Це дасть змогу досягнути кращого результату на наступних кроках.

Трансформація обличчя за допомогою face landmark estimation імплементована програмно за допомогою мови Python з використанням бібілотеки dlib подана нижче.

```
import sys
import dlib
import cv2
import openface

# Завантажуємо натреновану модель розпізнавання точок на обличчі:
predictor_model = "shape_predictor_68_face_landmarks.dat"

# Створюємо розпізнавача облич за допомогою HOG, який є вбудований в бібліотеці dlib
face_detector = dlib.get_frontal_face_detector()
face_pose_predictor = dlib.shape_predictor(predictor_model)
face_aligner = openface.AlignDlib(predictor_model)
```



```
# Дістаємо назву зображення з масиву даних командного рядка
file_name = sys.argv[1]
# Load the image
image = cv2.imread(file_name)

# Run the HOG face detector on the image data
detected_faces = face_detector(image, 1)

print("Found {} faces in the image file {}".format(len(detected_faces), file_name))

# У циклом опрацьовуємо всі обличчя, які були знайдені на зображенні
for i, face_rect in enumerate(detected_faces):

        # Знайдене обличчя повертається, як набір координат всіх 4 сторін
        print("- Face #{} found at Left: {} Top: {} Right: {} Bottom: {}".format(i, face_rect.left(),
face_rect.top(), face_rect.right(), face_rect.bottom()))
        # Отримати позицію лиця
        pose_landmarks = face_pose_predictor(image, face_rect)

        # Використовуючи openface обробляємо обличчя
        alignedFace = face_aligner.align(534, image, face_rect, landmarkIndices =
openface.AlignDlib.OUTER_EYES_AND_NOSE)

        # Зберегти оброблене зображення в файл jpg
        cv2.imwrite("aligned_face_{}.jpg".format(i), alignedFace)
```

Визначення унікальних особливостей обличчя

Цей етап безпосередньо пов’язаний із задачею розпізнавання. Найпростіший спосіб розпізнати обличчя – це порівнювати невідоме лице, яке отримано на другому кроці, зі всіма зображеннями облич, які є у тестовій базі даних зображень облич людей. Коли ми знаходимо подібне обличчя до невідомого, робимо висновок, що це є саме ця особа. Проте, коли тестова база має мільйони гігабайтів даних, необхідно переглянути всі наявні зображення і серед них вибрати те, яке подібне найбільше до заданого, то весь цей процес займатиме багато часу. Задача розпізнавання найчастіше використовують в таких сферах, де очікуваний час відповіді системи становить декілька секунд (наприклад, пошук особи з відеопотоку). Тому реалізація пошуку за алгоритмом перебору є неефективною через високу обчислювальну складність.

Для зменшення обчислювальної складності алгоритму запропоновано виділити декілька базових характеристик лиця. Відповідно, порівнюватимемо обличчя з наявними у тестовій базі за допомогою цих характеристик. Отже, виникає задача вибору таких характеристик, на основі яких комп’ютер буде розпізнавати обличчя. На перший погляд здається, що розмір або колір очей, довжина носу, величина губ, форма брів – це основні характеристики, за якими людина розпізнає обличчя. Проте,

як показали дослідження, ці характеристики, які для людей основні, для комп’ютерного розпізнавання не мають жодної цінності, оскільки аналіз зображення здійснюється попіксельно, а не загалом.

Для розв’язання цієї задачі запропоновано використовувати глибинно згорткову нейронну мережу (Deep Convolutional Neural Network), яка визначатиме 128 характеристик для кожного обличчя.

Процес навчання працює за принципом переглядання 3 зображень одночасно:

1. Завантажити зображення обличчя особи, яка є в тестовій базі даних (прокласифікованій).
2. Завантажити інше зображення обличчя цієї ж людини.
3. Завантажити зображення обличчя іншої людини.

Потім алгоритм аналізує характеристики, які він генерує для кожного з цих трьох зображень. Потім здійснюється навчання нейронної мережу так, щоб вона гарантувала, що характеристики і їх виміри, які мережа згенерувала для зображення №1 та №2, були якомога ближчі за значеннями, зображення №2 та №3 дальші. Ідея скорочувати складні необроблені дані в список комп’ютерно генерованих номерів набула найбільшого розвитку саме в машинному навчанні і вперше використовувалася у сфері перекладу. Підхід опрацювання облич таким чином, як запропоновано у статті, вперше був представлений інженерами компанії Google у 2015 році [4].



Рис. 4. Візуалізація роботи алгоритму

Процес тренування згорткової нейронної мережі для виділення цих особливих характеристик є трудоємним і вимагає наявності великої бази даних зображень облич і значної обчислювальної здатності комп’ютера. Навіть з використанням графічної карти NVidia Telsa, оскільки для більшої швидкості моделі тренують на відеокартах

компанії NVidia з наявністю підтримки CUDA, процес навчання займав приблизно 24 години [4] для отримання хороших результатів розпізнавання. Проте, як тільки нейронна мережа натренується, вона зможе генерувати характеристики для будь-якого обличчя, навіть для такого, яке вона ніколи не бачила до того. Отже, цей крок має бути здійснений тільки один раз, але від нього залежатиме успішність нейронної мережі у розпізнаванні облич. У разі, якщо немає змоги натренувати свою нейронну мережу, можна використати вже натреновану мережу [5]. У разі використання вже натренованої мережі, все що доведеться зробити, це передати зображення для розпізнавання 128 характеристик. Ось як це працює на тестовому зображенні:



Рис. 5. Отримано 128 характеристик для тестового зображення

Значення виділених характеристик важко інтерпретувати, для різних зображень частина з них може бути однакова, проте розташована у іншій послідовності.

Ідентифікація особи

Останній крок насправді є найпростішим кроком усього процесу. Все, що потрібно зробити, це знайти людину в базі даних відомих людей, яка найбільш подібна за характеристиками до виявленої людини на оброблюваному зображенні.

Реалізувати цей крок можна різними алгоритмами класифікації машинного навчання. Ми використовуватимемо лінійний класифікатор методу опорних векторів (SVM), який являє собою тренувальний набір даних з n точок вигляду [6]:

$$(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)$$

де y є 1 або -1, і кожен з них вказує клас, до якого належить точка $\vec{x}_i$. Кожен $\vec{x}_i$ є p-вимірним дійсним вектором. Нам треба знайти «максимально розділову гіперплощину», яка відділяє групу точок $\vec{x}_i$, для яких $y_i = 1$, від групи точок, для яких $y_i = -1$, і визначається таким чином, що відстань між цією гіперплощиною та найближчою точкою $\vec{x}_i$ з кожної з груп є максимальною.

Будь-яку гіперплощину може бути записано як множину точок $\vec{x}$, які задовольняють

$$\vec{w} * \vec{x} - b = 0$$

де $\vec{w}$ є (не обов'язково нормалізованим) вектором нормалі до цієї гіперплощини. Параметр $\frac{b}{\|\vec{w}\|}$ визначає зсув гіперплощини від початку координат вздовж вектора нормалі $\vec{w}$.

Усе, що потрібно зробити, це навчити SVM шукати найвдаліший збіг оброблюваного зображення із зображеннями з бази даних відомих облич. Час роботи цього класифікатора займає мілісекунди.

Висновки

Те, що технології розпізнавання облич надалі активно розвиватимуться, випливає з останніх конференцій, у яких приймали участь гіганти IT. Apple на останній конференції представила технологію розпізнавання облич Face ID. Ідея полягає в розблокуванні телефону за допомогою обличчя. Працює подібно до технології описаної у цій статті, тільки в їхній розробці є апаратна система визначення орієнтирів обличчя [7].Технології розпізнавання облич можуть бути затребувані у будь-якій сфері безпеки, бо якщо унікальність відбитка пальця є 1/100 000, то в обличчя це 1/100 000 000 [8]. Також значні кроки у сфері розпізнавання облич зробила соціальна мережа Facebook. Після завантаження нової фотографії і позначення на ній обличчя людини, вже за деякий час, позначатиме її автоматично на новому фото, яке ви готуєте до публікації. Відсоток правильності позначень за підрахунками компанії Facebook складає 98% [9]. Тобто, ринок цієї технології невпинно зростатиме протягом наступних 5-10 років. У статті описано та скомбіновано базові алгоритми для роботи із зображеннями та розпізнавання облич.

Література

1. Li S.Z. Handbook of Face Recognition / S.Z. Li, A. Jain; – M.: Springer New York, 2011. – 398 p.
2. Histograms of Oriented Gradients for Human Detection // [Електр. Ресурс]. – Режим доступу: https://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf
3. Theodoridis S. Pattern Recognition 4th Edition / S. Theodoridis, K. Koutroumbas; - M.: Academic Press, 2008. – 981 p.
4. FaceNet: A Unified Embedding for Face Recognition and Clustering // [Електр. Ресурс]. – Режим доступу: https://www.cv-foundation.org/openaccess/content_cvpr_2015 /app/1A_089.pdf
5. OpenFace. Free and open source face recognition with deep neural networks // [Електр. Ресурс]. – Режим доступу: https://cmusatyalab.github.io/openface/
6. Ma Yu. Support Vector Machines Applications / Yu. Ma, G. Guo; – M.: Springer Science & Business Media, 2014. – 302 p.
7. Apple Special Event // [Електр. Ресурс]. – Режим доступу: https://www.apple.com/apple-events/september-2017/.
8. Face ID // [Електр. Ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/Face_ID
9. DeepFace: Closing the Gap to Human-Level Performance in Face Verification // [Електр. Ресурс]. – Режим доступу: https://research.fb.com/publications/deepface-closing-the-gap-to-human-level-performance-in-face-verification/

RESUME

N.B. Shakhovska, O.A. Basystiuk

Face recognition based on machine learning algorithms

Face recognition is complicated problem. For example, problems like predicting house price, telling if image contains an object, creating new data based on existing data, is problems that can be solved by choosing one machine learning algorithm, feeding in this algorithm data and getting the result. But face recognition contains under the hood series of related problems:

1. Find face at picture.
2. Analyze all faces, to understand if the face is turned.
3. Pick unique features of the face.
4. Compare this feature set with dataset of persons that you already know.

Therefore, purpose of the work is to develop pipeline in which we will solve by machine learning algorithms each problem and feed result data to the next step. In this paper is proposed usage of histogram of oriented gradients(HOG) for analysing of the picture and finding faces on it, because this method ignore problems of bad lighting at the

picture. After getting HOG data it will be normalize by using L2-norm algorithm, because it gives us most accurate data and increase performance of face recognition process. For creating face landmark is proposed face landmarks estimation, this step gives possibility to alter picture for centered face as best as possible. Finally, 128 unique features of face will be picked by pre-trained deep convolutional neural network and compare with already known by support vector machine. Results of research can be used in all types of face recognition problem fields.

Face recognition problem is actively researchable nowadays, there are solution created by Apple, called Face ID, the main goal of this is unlock the phone by face. Facebook algorithm help you to target your friends at your new released photo. In my opinion, face recognition in future will become a significant part in our life. This technology has huge possibilities in security and entertainment field. This paper helps you understand basic techniques of face recognition and create your own face recognition software.

Надійшла до редакції 06.10.2017